

SE : GESTION DES FICHIERS ET RÉPERTOIRES, DROITS D'ACCÈS

OBJECTIFS

1. *Comprendre les notions de :*
 - *système d'exploitation Unix Linux*
 - *terminal ou shell*
 - *fichiers et organisation arborescente*
2. *Savoir naviguer dans l'arborescence :*
 - *Racine / et raccourcis.,*
 - Chemin de fichier (filepath) relatif et absolu*
 - *commandes ls, cd, pwd....*
3. *maîtriser les droits d'accès aux fichiers :*
 - *afficher des info. concernant utilisateurs et groupes*
 - *afficher des droits fichiers par fichier*
 - *modification des droits via un code numérique ou les caractères spéciaux*
 - *modification du propriétaire ou de l'appartenance un groupe*

BON SAVOIR (EN MODE CONSOLE)

- On ne tape jamais les commandes ou les chemins en entier, on utilise la **touche TAB** (tabulation) pour l'auto-complétion
- Pour remonter dans les commandes précédentes, on peut utiliser la touche **flèche haut** ↑ du clavier
- La commande `history` permet d'obtenir l'historique de toutes les commandes précédemment tapées
- Depuis l'interface graphique, on peut passer en **mode console** ? Pour cela, il faut appuyer simultanément sur **CTRL+ALT+F1** ou F2 → F6. **CTRL+ALT+F7** permet de revenir au mode graphique.
- `clear` ou la combinaison **CTRL + I** permet de vider la console des commandes précédemment rentrées
- **CTRL + R** permet d'activer le *reverse-i-search*, c'est à dire la possibilité de chercher dans d'anciennes commandes en saisissant une chaîne de caractères... Très efficace !
- Sur un environnement Mate, on peut utiliser la **touche F12** pour faire apparaître un terminal.
- La commande **man** suivie d'un argument permet d'accéder la documentation détaillée de la commande passée en argument (exemple : `man ls`).

SYSTÈME D 'EXPLOITATION UNIX LINUX

Rappel : Le système d'exploitation est le logiciel central d'un appareil informatique. Il se place entre

le matériel et les logiciels applicatifs. Ses principales fonctions sont : le contrôle du matériel, la gestion des logiciels installés et leur exécution, l'organisation des données, la communication avec l'utilisateur via une interface homme-machine dont le type peut varier selon les systèmes (interface graphique, mode fenêtré, interface textuelle, avec menus déroulants ou bien interpréteur, etc.).

Tout système d'exploitation s'intègre dans ce qu'on appelle un modèle en couches, qui permet une abstraction progressive par rapport aux spécificités du matériel. Un exemple de système d'exploitation très utilisé par les particuliers est Windows 7. Comme les systèmes évoluent avec le temps, il existe de nombreuses versions différentes. Parfois il est plus pratique de parler de familles de systèmes. Par exemple Windows 95, Windows XP, Windows 7 appartiennent tous la famille des systèmes Windows. De même, il existe des systèmes

d'exploitation tels que Ubuntu, Mandriva qui appartiennent tous la famille de systèmes Unix. c'est avec un tel système d'exploitation que nous travaillerons.

Les caractéristiques de systèmes Unix est qu'ils sont multi-utilisateurs et multitâches. Un ensemble de fichiers importants existent sur tout système Unix, nous en donnons ici une liste non-exhaustive.

bin : commandes de base de l'utilisateur

./dev : fichiers spéciaux (correspondent aux périphériques

./etc : commandes et fichiers d'administration (fichiers de démarrage)

/lib : routines compilées

tmp : fichiers de travail, temporaires

/usr/bin : commandes complémentaires de l'utilisateur

/usr/include : fichiers d'en-tête des langages

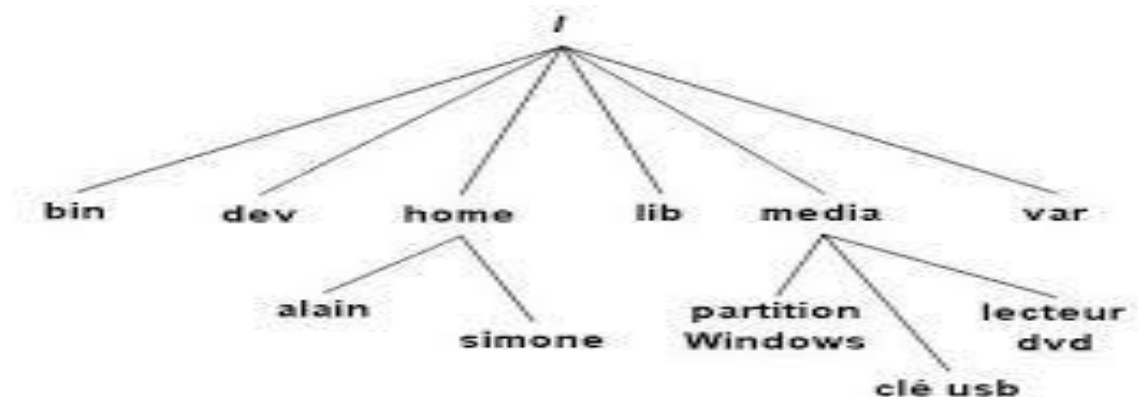
home : fichiers des utilisateurs

/sbin et /usr/sbin : commandes pour le super-utilisateur root

var : contient des données diverses, notamment des historiques

FICHIERS ET ORGANISATION ARBORESCENTE

Sous Unix, «tout est fichier Cela signifie (111 'un disque, une mémoire, un lecteur de disquette, la sortie d'un processus sont tous vus par le système comme des fichiers. Il est donc important que ces fichiers sont bien ordonnés. Un système de fichiers sous Unix (aussi appelé ext2) est organisé de manière arborescente comme le montre le schéma ci-dessous.



En cliquant sur Ordinateur ou Système de fichiers dans Nautlius, on trouve toute l'arborescence du système de fichiers. Cela paraît compliqué quand on vient d'un système Windows© mais il faut comprendre qu'un système Gnu/Linux est construit quasi exclusivement avec des fichiers textes : un périphérique (port USB, carte son...), un programme en cours d'exécution, une partition, un fichier sont des fichiers textes !

Quasiment tous les fichiers de configuration du système sont éditables avec un éditeur de texte (Gedit sur Gnome, Pluma sur Mate, Mousepad sur Xfce...).

Signalons quelques dossiers :

- **bin** : programmes utilisés par les utilisateurs et les administrateurs du système (rmdir, mkdir...)
- **boot** : fichiers de démarrage du système (noyau du système, etc.). On y trouve aussi certains fichiers de configuration (GRUB...)
- **sbin** : programmes systèmes et outils d'administration (formatage d'un disque dur...)
- **dev** : comme « devices » ; il contient tous les périphériques matériels
- **etc** : tous les fichiers de configuration. Ce sont des fichiers textes qu'on peut donc modifier, à condition de les éditer en sudo (exemple : hostname)
- **home** : contient les données des utilisateurs et les paramètres de configuration utilisateurs des logiciels. On fera en sorte que ce dossier soit déporté sur une autre partition !
- **lib** : contient les principales bibliothèques partagées (équivalent des DLL de Windows).
- **media** : C'est dans ce dossier que se montent naturellement tous les périphériques externes. Ils apparaissent alors dans le panneau latéral de Nautilus.
- **mnt** : pour monter les périphériques ailleurs que dans media. Dans ce cas, ils n'apparaissent pas dans le panneau latéral de Nautilus.
- **usr** : Pour Unix System Ressources. C'est là qu'on trouve la grande majorité des applications installées
- **var** : on y trouve entre autre les fichiers de log. Mais aussi souvent les applications web (dans /var/www/).

CHEMIN DE FICHIER RELATIF/ABSOLU

La racine est un répertoire particulier, en ceci qu'il est le tout premier niveau de l'arborescence. Le symbole pour la racine est Votre répertoire principal, celui on vous vous situez initialement en ouvrant un terminal peut-être abrégé par le caractère . et sont deux répertoires particuliers : . référence le dossier courant et le répertoire parent.

Chaque répertoire (ou chaque fichier) a un identifiant unique; un «chemin absolu»dans l' arborescence. Cette désignation du répertoire ou du fichier décrit le chemin partant de la racine pour arriver au répertoire ou fichier. Un chemin absolu commence donc toujours par un /

Par ailleurs, il est possible de nommer un répertoire ou un fichier partir de l'endroit courant. On appelle cela un chemin relatif

Voyons quelques exemples :

Chemin absolu : ./etc, / users/imll, .

chemins relatifs (depuis votre répertoire personnel) : Bureau, . Documents, .././prof

NOTION DE SUDO

Par défaut, le compte super utilisateur (root) n'a pas de mot de passe sous Ubuntu. Sous les systèmes Unix (dont Linux fait partie), on ne peut pas se connecter directement à un compte utilisateur sans mot de passe. Ceci signifie donc que vous ne pouvez pas vous connecter en tant que root.

Mais alors, comment faire pour exécuter des commandes qui nécessitent les privilèges administrateur (souvent des tâches administratives et sensibles, pouvant corrompre le système) ?

Pour cela, il faut utiliser l'utilitaire **sudo** (Substitute User DO).

Par défaut, le premier utilisateur, celui qui a été créé lors de l'installation du système, peut exécuter des tâches d'administration grâce à **sudo**. Le mot de passe demandé est celui de l'utilisateur courant. La commande sera exécutée si le mot de passe entré est correct et que l'utilisateur courant peut effectuer des tâches d'administration.

L'utilitaire **sudo** s'utilise en ligne de commande, dans un terminal. Il sert à exécuter, en mode superutilisateur, des commandes ou des applications en console.

Pour lancer des applications graphiques avec les privilèges d'administration, il faut utiliser les pendants graphiques comme **gksudo** (dans les faits, c'est extrêmement rare d'utiliser **gksudo**).

QUAND UTILISONS-NOUS SUDO ?

sudo sera utile uniquement pour exécuter des tâches d'administration : action sur le système de fichiers, création de comptes utilisateurs, installation d'applications...

Il n'est pas utile pour agir dans son répertoire personnel stocké dans /home/login/

Pour plus d'informations concernant l'utilisation et le paramétrage de **sudo** : <http://doc.ubuntu-fr.org/sudo>

SE DÉPLACER, CRÉER DES FICHIERS DANS DES DOSSIERS...

Commande	Action	Exemples	Remarques
ls	lister le contenu d'un dossier	<code>ls -l Documents/</code>	<ul style="list-style-type: none">-l permet un affichage détaillé-a de voir les fichiers cachésSur certains systèmes, <code>ll</code> est un alias de <code>ls -al</code> (Cf. le fichier <code>.bashrc</code>)
cd	se déplacer dans l'arborescence	<code>cd /var/log/</code>	<ul style="list-style-type: none"><code>cd</code> pour revenir au dossier utilisateur<code>cd ..</code> pour remonter d'un cran
mv	déplacer ou renommer	<code>mv test.html essai.html</code>	<ul style="list-style-type: none">-f : force l'exécution
cp	copier des fichiers ou des dossiers	<code>cp fichier1.html rep2/.</code>	<ul style="list-style-type: none">-r : copie les sous dossiers
rm	supprimer des fichiers	<code>rm -f fichier.html</code>	<ul style="list-style-type: none">-f : force l'exécution , pas de confirmation-r : permet de supprimer un dossier
mkdir	créer un dossier	<code>mkdir -p test/essai</code>	<ul style="list-style-type: none">-p : crée les dossiers parents s'ils n'existent pas
rmdir	supprimer un dossier vide	<code>rmdir -p test/essai</code>	<ul style="list-style-type: none">-p : supprime le dossier parent s'il est vide

AFFICHER DES FICHIERS, LISTER DU CONTENU...

Commande	Action	Exemples	Remarques
----------	--------	----------	-----------

cat	lister le contenu d'un fichier texte	cat /etc/fichier	
nano	éditer un fichier texte	nano /etc/fichier	<ul style="list-style-type: none"> • Plus simple pour les débutants que vi ou vim • Les manipulations se font avec une combinaison de touches CTRL + ... • bien entendu, on peut éditer un fichier dans l'interface graphique avec l'éditeur gedit (ou pluma)
vi / vim	éditer un fichier texte dans une console	vim /etc/fichier	<ul style="list-style-type: none"> • vim n'est pas installé par défaut • i pour passer en insertion • echap puis :wq pour enregistrer (w) et quitter (q)
head	visualise le haut du fichier	head /etc/fichier	<ul style="list-style-type: none"> • l'argument -n 100 affiche les 100 premières lignes
tail	visualise le bas du fichier	tail /etc/fichier	<ul style="list-style-type: none"> • l'argument -n 100 affiche les 100 dernières lignes
less	parcourir facilement un fichier avec les flèches et pageUp, pageDown	less /var/log/syslog	<ul style="list-style-type: none"> • On sort en tapant :q

FAIRE DES RECHERCHES...

Commande	Action	Exemples	Remarques
grep	rechercher une chaîne de caractères dans un fichier	grep systemd /etc/passwd	<ul style="list-style-type: none"> • Pour une recherche récursive, on peut utiliser rgrep
locate	rechercher des fichiers	locate fichier	<ul style="list-style-type: none"> • Trouve très rapidement un fichier à condition qu'il ait été indexé
find	chercher des fichiers et éventuellement d'exécuter des commandes sur ceux-ci	<p>Trouver tous les fichiers mp3 dans son /home :</p> <ul style="list-style-type: none"> • find /home/\$USER -type f -name "*.mp3" <p>Les supprimer...</p> <ul style="list-style-type: none"> • find / -type f -name "autorun.*" -exec rm {} \; <p>Trouver des fichiers de plus de 500 Mo dans / et crée un fichier qui liste ces fichiers :</p> <ul style="list-style-type: none"> • find / -type f -size +500000k -exec ls -al {} \; > liste_500.log 	<p>Options les plus fréquentes :</p> <ul style="list-style-type: none"> • -name : Recherche d'un fichier par son nom • -iname : Même chose que -name mais insensible à la casse • -type : Recherche de fichier d'un certain type • -atime : Recherche par date de dernier accès • -mtime : Recherche par date de dernière modification • -link : Recherche du nombre de liens au fichier • -user : Recherche de fichiers appartenant à l'utilisateur donné

			<ul style="list-style-type: none"> • -group : Recherche de fichiers appartenant au groupe donné <p>Actions les plus fréquentes :</p> <ul style="list-style-type: none"> • -exec : Exécute la commande donnée aux fichiers trouvés • -ok : Même chose que -exec mais demande une confirmation • -ls : exécute la commande ls à chaque fichier trouvé
--	--	--	--

UTILISATEURS / GROUPES/ DROITS D'ACCÈS

AFFICHAGE DES INFORMATIONS UTILISATEURS ET GROUPES

Rappel : Les systèmes Unix sont dits multi-utilisateur, cela signifie que différentes personnes peuvent utiliser un ordinateur sans que cela pose un problème. Chaque fichier attribué par un utilisateur, ce qui permet de mettre en place des privilèges et des restrictions, pour des raisons la fois de sécurité et de confidentialité. On peut aussi attribuer des privilèges un ensemble d'utilisateurs autres que le propriétaire l'aide des "groupes".

L'information globale sur les comptes utilisateurs est stockée dans le fichier /etc/passwd (consulté lors de la saisie du mot de passe). Le format de chaque ligne de ce fichier spécial est le suivant :

- *nom :passe :uid :gid :commentaire :répertoire natif :shell*
- *uid : user identifier*
- *gid : group identifier*

AFFICHAGE DES DROITS D'ACCÈS

Tout fichier présent sur le système possède un certain nombre d'attributs définissant les droits d'accès ce fichier pour les utilisateurs du système. Pour définir ces attributs, on se base sur la définition de trois classes d'utilisateurs : "le propriétaire", "le groupe" et "les autres". A ces trois classes d'utilisateurs, on peut donner trois types de permissions sur un fichier : "lecture" "écriture" (w) et "exécution" (x). Quand le droit n'est pas accordé, il y a un la place de la permission. Lorsque vous avez utilisé la commande 'ls -al', vous avez pu voir les différents droits des fichiers et répertoires de votre répertoire de travail. La ligne affichée par 'ls' commence avec le codage de ces droits pour chaque fichier. Tout d'abord, un premier caractère est réservé pour indiquer un type de fichiers particulier ('l' pour lien, 'd' pour répertoire), et ensuite trois caractères pour les droits du "propriétaire", suivi par trois caractères pour les droits du "groupe" du propriétaire et enfin, trois caractères pour les droits des "autres" utilisateurs (ceux qui ne font pas partie du groupe du propriétaire). Lorsque le caractère est, c'est que le droit de lire, écrire ou exécuter n'est pas accordé.

Exemple : -rwxrw-r- nous avons un fichier dont le propriétaire a tous les droits (rwx), le groupe peut seulement lire et écrire (rw-) et les autres ne peuvent que le lire

MODIFICATION DES DROITS D'ACCÈS

Si vous voulez donner le droit de modifier ou d'exécuter vos fichiers aux autres, vous pouvez le faire grâce la commande '**chmod**'. La représentation de ces droits dans la commande '**chmod**' est codifié. "r" est codé par le chiffre 4, "w" par le chiffre 2 et "x" par le chiffre 1. Avec cette représentation on peut donner les droits d'un fichier avec 3 chiffres qui correspondent aux droits du propriétaire, du groupe et des autres. Exemple :
propriétaire (tous les droits) : rwx (4+2+1) donne 7 groupes (lecture et écriture) rw- (4+2) donne 6 autres (lecture et exécution) : r-x (4+1) donne 5 Donc en respectant l'ordre expliqué dans l'exercice précédent, nous avons "765" Donc la commande correspondante sera "**chmod 765 fichier**".

MODIFICATION DU PROPRIÉTAIRE OU DES DROITS D'ACCÈS

Vous pouvez également attribuer l'appartenance d'un fichier un autre utilisateur du système avec '**chown**', et le faire appartenir un autre groupe avec '**chgrp**'. Essayez ces commandes sur deux des fichiers textes précédemment utilisés pour les attribuer quelqu'un d'autre et un autre groupe que celui par défaut. Pouvez-vous encore les modifier ? Les supprimer ? Pourquoi ?